

Programa del curso IC-3101

Arquitectura de Computadoras

Escuela de Computación
Carrera de Ingeniería en Computación, Plan 410.

2 Descripción general En este curso se estudiarán todos aquellos recursos de la computadora que son visibles al programador de bajo nivel. El estudiante será preparado en programación en lenguaje ensamblador para una arquitectura particular, aunque al mismo tiempo se estudian conceptos arquitectónicos presentes de manera extendida en computadoras actuales

3 Objetivos

Objetivo General
Analizar el conocimiento sobre una arquitectura de computadoras actual para la programación desde una perspectiva de bajo nivel.

Objetivos Específicos

1. Programar de manera competente en el lenguaje ensamblador de al menos una arquitectura.
2. Entender los alcances y limitaciones que una arquitectura establece a un potencial programador.
3. Dominar conceptos universales de Arquitectura de Computadoras presentes en diversas computadoras actuales.
4. Comprender el funcionamiento general de sistemas empotrados

4 Contenidos

1. Introducción

- 1.1.** Organización vs. Arquitectura de Computadoras
- 1.2.** Conceptos generales y Repaso histórico
- 1.3.** Medidas de rendimiento
- 1.4.** ¿Por qué aprender ensamblador?
- 1.5.** Arquitectura ejemplo

2. Infraestructura de software

- 2.1.** Ensamblador
- 2.2.** Linker
- 2.3.** Loader
- 2.4.** Lenguajes de Alto nivel
- 2.5.** Simuladores
- 2.6.** Relación con Sistema Operativo

3. Aritmética para computadoras

- 3.1.** Repaso Sistemas Numéricos
- 3.2.** Diseño general de una ALU
- 3.3.** Enteros con y sin signo
- 3.4.** Suma, resta y operaciones lógicas
- 3.5.** Multiplicación y División
- 3.6.** Punto Flotante
- 3.7.** Aproximaciones

4. Lenguaje Ensamblador

- 4.1.** Conceptos introductorios.
- 4.2.** Formatos de Instrucción.
- 4.3.** Modos de Direccionamiento.
- 4.4.** Conjunto de instrucciones básico.
- 4.5.** Proceso de ensamblaje - desensamblaje.
- 4.6.** Programación básica con el ensamblador (aritmética y flujo de control).
- 4.7.** Paso de parámetros y construcción de rutinas.
- 4.8.** Macros y preensamblaje.
- 4.9.** Directivas y Operadores avanzados.
- 4.10.** Modularización, "Linking" y "Loading"
- 4.11.** Formatos de módulos ejecutables
- 4.12.** Comunicación entre Programas

5. Pipeline y modelos de alto rendimiento

- 5.1.** Introducción y conceptos básicos
- 5.2.** Paralelismo vs concurrencia
- 5.3.** Pipeline conceptos y dificultades de implantación
- 5.4.** Pipeline de instrucciones
- 5.5.** Predicción de bifurcaciones
- 5.6.** Otros modelos de alto rendimiento

6. RISC vs CISC

- 6.1.** Diferencias y características de ejecución
- 6.2.** Ventajas de CISC y Ventajas de RISC

7. Entrada/Salida

- 7.1.** Conceptos básicos

- 7.2.** Tipos y características de dispositivos de E/S
- 7.3.** Interfaces E/S con memoria, CPU y Sistema Operativo
- 7.4.** DMA – E/S mapeada a memoria
- 7.5.** Diseño de Sistemas de E/S

8. Multiprocesadores

- 8.1.** Introducción
- 8.2.** Conceptos de Programación
- 8.3.** Organizaciones
- 8.4.** Interconection Network
- 8.5.** Sistemas Multicore
- 8.6.** Threads de hardware
- 8.7.** Multiprocesadores y jerarquía de memorias
- 8.8.** Protocolos de coherencia

9. Sistemas empotrados

- 9.1.** Definiciones
- 9.2.** System-on-chip
- 9.3.** Aplicaciones

**5 Metodología
de enseñanza
y aprendizaje**

Clases magistrales donde el profesor desarrollará el material teórico asociado al curso. Se seguirán dos grandes ramas: entrenamiento en programación ensamblador y conceptos de arquitectura de computadoras. Se recomienda estos dos temas que no sean cubiertos de manera secuencial, sino de manera intercalada para una mejor comprensión e ilustración de los temas aprendidos en cada rama. Los estudiantes deberán estudiar independientemente material adicional asignado en la forma de lecturas o investigaciones bibliográficas, y desarrollarán una serie de proyectos de programación en lenguaje ensamblador usando preferiblemente equipos instalados en nuestros laboratorios o en su defecto recurriendo a simuladores de software de las arquitecturas estudiadas. En lo posible se contrastará el ensamblador ejemplo con algún otro ensamblador que permita apreciar diferencias sustanciales entre ellos.

6 Evaluación

Rubro	Porcentaje
Proyectos de investigación	20%
Proyectos Programados en Ensamblador	80%
Total	100%

Cronograma de Actividades

Actividad	Semanas
Introducción	1
Infraestructura de software	0.5
Aritmética para computadoras	2
Lenguaje Ensamblador	6
Pipeline y modelos de alto rendimiento	2
RISC vs CISC	0.5
Entrada/Salida	1
Multiprocesadores	2
Sistemas empotrados	1

7 Bibliografía

Obligatoria

Rauber, T., & Rünger, G. (2010). *Parallel Programming: for Multicore and Cluster Systems*. Alemania: Springer-Verlag.

Englander, I. (2002). *Arquitectura computacional* (1era edición ed.). México: Compañía editorial continental.

Furber, S. (2000). *ARM System-on-Chip Architecture*. Pearson Education.

Hennessy, J., & Patterson, D. (2012). *Computer Architecture: A Quantitative Approach* (5ta. Edición ed.). Morgan Kauffman.

Hohl, W. (2009). *ARM Assembly Language*. CRC Press.

Stallings, W. (2006). *Organización y Arquitectura de Computadores*

(7ma edición ed.). México: Pearson Educación.

Patterson, D., & Hennessy, J. (2009). *Computer Organization & Design* (4ta. Edición ed.). Morgan Kauffman.

Irving, K. (2008). *Lenguaje Ensamblador para computadoras basadas en Intel* (5ta edición ed.). México: Pearson Prentice Hall.

Duran, L. (2007). *El gran libro del PC Interno* (1era edición ed.). Barcelona: Grupo Marcombo ediciones técnicas.

Tanenbaum, A. (2005). *Structured Computer Organization* (5ta. Edición ed.). Prentice Hall.

Adicional

No tiene Bibliografía adicional.

8 Profesor

Jorge A. Vargas Calvo
Oficina: 19, Escuela de Computación
Teléfono (oficina, ext.): 2472
Correo electrónico: avargas@itcr.ac.cr; jorgevargas.cursostec@gmail.com
Horas de consulta: Martes 1:00pm a 4:45 pm